REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1999	Final 6/24/97-9/23/98

4. TITLE AND SUBTITLE

Efficient Adaptive Optical Processing for Sonar Arrays

5. FUNDING NUMBERS

N00014-97-C-2039

6. AUTHOR(S)

Kelvin Wagner, Shawn Kraut, Lloyd Griffiths

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)

University of Colorado
Optoelectronic Computing Systems Center
Campus Box 525
Boulder, CO 80309-0425

8. PERFORMING ORGANIZATION REPORT NUMBER

153-6925

9. SPONSORING / MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

Dr. John Lee
Naval Research Laboratory
Washington, DC 20375-5338

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

a. DISTRIBUTION / AVAILABILITY STATEMENT

Unlimited

12. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In this final report we summarize our progress towards developing an innovative algorithm called BEAMTAP (Broadband Efficient Adaptive Method for True-time-delay Array Processing) for adaptive array processing in broadband scenarios. This approach is especially appropriate for optical implementation of sonar array beamforming. The BEAMTAP algorithm only requires two tapped delay lines, rather than one for every array element as required by conventional time-delay-and-sum beamformers. This efficiency provides a huge hardware savings for large arrays. By separating the delay lines from the adaptive weights, this algorithm can be more easily implemented in optical hardware, making it easier to exploit processing and interfacing advantages of optical systems. We discuss an optical beamforming and jammer-nulling system for sonar array beamforming that utilizes photorefractive crystals for the adaptive weights in combination with time-delay-and-integrate (TDI) charge-coupled device (CCD) detector arrays to process the parallel data from optically interrogated arrays of hydrophone sensors. In this report we discuss the operation of BEAMTAP in comparison with LMS time-delay-and-sum array processors and P-vector passive array beamsteering operations, and we demonstrate its operation using numerical simulations.

14. SUBJECT TERMS

Adaptive array processing, sonar, photorefractive crystals, optical processing

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Efficient Adaptive Optical Processing for Sonar Arrays

K. Wagner, S. Kraut
Dept of ECE
box 425
University of Colorado
Boulder CO 80309-0425
Phone : (303)492-4661
FAX : (303)492-5810
email : kelvin@colorado.edu

L. Griffiths, Dean
School of Information Technology and Engineering
100 Science and Technology II, ms 4A3
George Mason University
Fairfax VA 22030-4444

Program Manager: **Dr. John Lee**
code 5623
Naval Research Lab
office b216/200m
4555 Overlook Ave., SW
(202)-767-3100
lee8@ccf.nrl.navy.mil

Original Period of Performance: (Phase 1) June 24 1997 – Mar 23 1998
6 month No cost extension Mar 24 1998 – Sept 23 1999
(Phase 2) Mar 24 1998 – Mar 24 1999 (not exercised)

Date of report April 23 1999

1 Abstract

In this final report we summarize our progress towards developing an innovative algorithm called BEAMTAP (Broadband Efficient Adaptive Method for True-time-delay Array Processing) for adaptive array processing in broadband scenarios. This approach is especially appropriate for optical implementation of sonar array beamforming. The BEAMTAP algorithm only requires two tapped delay lines, rather than one for every array element as required by conventional time-delay-and-sum beamformers. This efficiency provides a huge

hardware savings for large arrays. By separating the delay lines from the adaptive weights, this algorithm can be more easily implemented in optical hardware, making it easier to exploit processing and interfacing advantages of optical systems. We discuss an optical beamforming and jammer-nulling system for sonar array beamforming that utilizes photorefractive crystals for the adaptive weights in combination with time-delay-and-integrate (TDI) charge-coupled device (CCD) detector arrays to process the parallel data from optically interrogated arrays of hydrophone sensors. In this report we discuss the operation of BEAMTAP in comparison with LMS time-delay-and-sum array processors and P-vector passive array beamsteering operations, and we demonstrate its operation using numerical simulations.

2 Executive Summary

In this project we investigated à novel algorithm called BEAMTAP (Broaband Efficient Adaptive Method for True-time-delay Array Processing) for efficient adaptive processing of data from sonar arrays that is uniquely suited to real time optical implementation. This BEAMTAP architecture has a compact mapping into optical hardware, allowing for efficient interfacing with front end interferometric optical hydrophone sonar sensors. Massively parallel optical implementation also allows for real-time, fully broadband adaptation of arrays with thousands of elements. In addition, the use of real-time holographic materials, such as photorefractives, will enable the system to continually readapt as the signal environment changes. The key to this approach is a real-time holographic media (photorefractive crystal) that stores adaptive parameters (or weights) as holographic gratings and allows them to be continually readapted as the signal environment changes. But because there are no intrinsic

time delay on the scale needed for sonar (milliseconds) within the weights, the conventional time-delay-and-sum approach must be modified, and this has led us to the BEAMTAP architecture.

Broadband true-time-delay (TTD) processing is essential in applications such as sonar processing, where the bandwidth of sources can easily be large compared to their center frequency. Without suitable broadband processing, sources that emit a range of temporal frequencies (eg motors or propellors) may appear to be coming from a range of angles, preventing them from being well discriminated from other sources. Broadband processing requires a huge increase of the number of adaptive parameters required for narrowband processing, making optical parallel processing even more advantageous in broadband scenarios.

Until now broadband beamforming in the optical domain has largely been based on optically delaying the signals from the array elements. This requires a tapped delay line (TDL) for every array element, involving a huge amount of hardware and inefficient use of space. The novel approach detailed in this report, BEAMTAP (for Broadband Efficient Adaptive Method for Time-delay Array Processing), replaces the 1000 or more TDLs with only two delay lines: a tap-out scrolling input modulator, and a tap-in time-delay-and-integrate (TDI) output detector. The calculated adaptive weights can now be compactly contained in less than a cubic centimeter of space in a single photorefractive crystal, reducing the costs associated with space, hardware, and power consumption.

This architecture is equally appropriate for active and passive listening scenarios. Using the emitted waveform (or sonar ping) in active listening would allow determination of source location and Doppler shift with no a priori knowledge of the array geometry. Passive listening can be used to greatly enhance the signal coming from a direction of interest, optimally

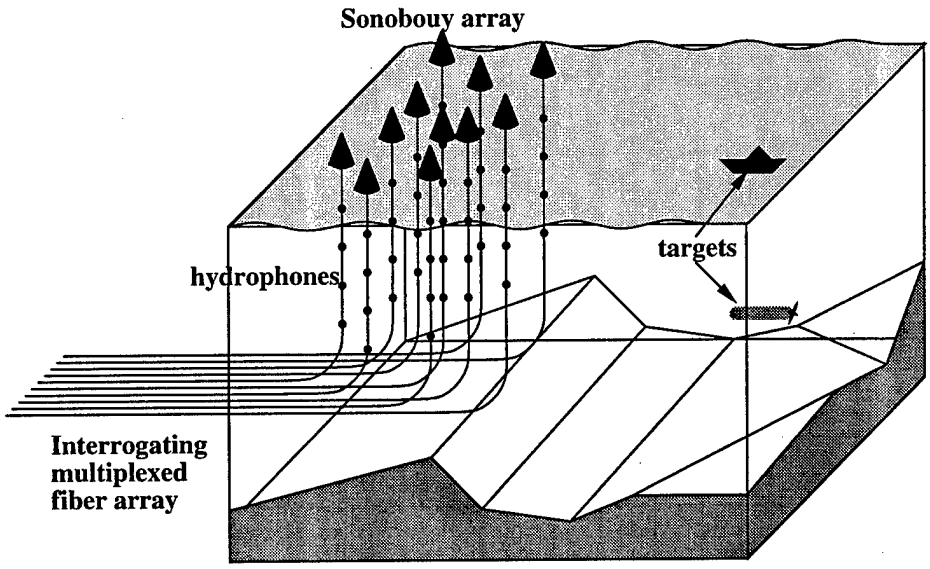

Figure 1: The passive sonobuoy array environment.

suppressing noise and other interfering sources. The output signal can then be passed on for subsequent post-processing, either optical or conventional electronic digital signal processing, such as source identification and recognition.

The many advantages of optical processing include faster computation, real-time adaptation, and interfacing with front-end optical sensors and optical post-processors. By providing an optical implementation that is power efficient, hardware efficient, and spatially compact, BEAMTAP makes possible real-time adaptation that utilizes the full adaptive degrees of freedom, providing an optimally filtered temporal signal that can be further processed for waveform classification and target identification.

3 Statement of the Problem

The underwater array sensing, beamforming, imaging, and target classification problem using towed, fixed, or sonobouy hydrophone arrays of fiber multiplexed optical sensors (Figure 1) is extremely challenging. In this report we present a real-time optical processing approach that exploits the optical nature of the data emerging from the arrays of fibers to directly process the received acoustic signals, allowing adaptive beam-forming and imaging in a compact power-efficient optical processor. This processing task requires compatibility between the sensors, the fiber multiplexing technique (eg. time, frequency, wavelength, coherence), the demultiplexing and detection of the sensor waveforms, the capabilities, resolution, and response time of the optical processor, and the algorithmic and mission requirements of the underwater sensing application. Optical processing allows for efficient coupling to front-end optically based sensors, and provides output appropriate for optical post-processing.

The many well known difficulties encountered in both the passive and active sonar signal processing environment include the following: the multiple decade sonar bandwidth which requires broadband true-time-delay (TTD) processing, inhomogeneities and waveguiding of the propagation medium, numerous multipath and echoes, especially in shallows, highly dispersive media, frequency dependent absorption, interference and noise sources that must be cancelled, significant broadband target Doppler shifts, and poorly sampled and drifting arrays. Fully adaptive systems may be able to deal with most of these difficulties, but are usually impractical due to the tremendous computational burdens illustrated by the following example. Consider an array of $m_1 = 100$ sonobuoys randomly scattered over a square kilometer, each with $m_2 = 100$ optical hydrophones vertically distributed throughout a 100 m depth, as illustrated in Figure 1. The hydrophone sensors in each sonobouy array are optically interrogated by the modulated optical signals and multiplexed in time or wavelength

onto a single fiber, producing an array of 100 fibers that are transmitted back to the processor. All of the m_2 signals must be demultiplexed from the m_1 fibers using a parallel array of coherent fiber receivers, producing $m = m_1 m_2$ total measured signals that must be processed. Time delays of at least 1 second may be required in the processing due to the 1.5 km maximum diagonal propagation distance across the array, and with 10 - 1000 Hz bandwidth sampled at 1 Hz resolution, at least $l = 1000$ time delays of each signal will be required, leading to a requirement for $N = ml = 10^7$ adaptive weights. Since optimal, fully adaptive processing typically involves complexity equivalent to covariance matrix estimation and inversion, this leads to storage requirements ($N^2 = 10^{14}$) in excess of Terabytes, and since updates are required about once every second, processing throughput approaching N^3 operations/sec is required. Furthermore, fast constrained algorithms require between $m^2 l^2 = 10^{14}$ (iterative inversion) and $m^3 l = 10^{15}$ (block Toeplitz inversion) operations per sample, yielding astronomical processing throughputs approaching 10^{18} ops/sec. These processing loads are well beyond even the most optimistic projections for massively parallel advanced wafer scale and multi-chip module DSP implementations, which are only projected to approach petaflop rates well into the next century —still a factor of 1000 too slow for fully adaptive large sonar array requirements. Since new targets can appear within a time frame of tens of seconds, and the array can drift to a new configuration requiring readaptation in this same time scale, efficient perturbative approaches may not be effective at reducing these required throughputs. This is the reason that most DSP approaches utilize techniques that reduce the problem size, such as subarray adaptation. In this way the number of adaptive parameters are reduced at the expense of sacrificing optimal processing, thus diminishing their ability to process data from complex signal environments with large numbers of targets.

As an alternative, in this proiject we investigated a class of optical processors that we have previously employed [1, 2, 3, 4] for adaptive phased-array radar jammer-nulling and beam-steering operations. This architecture is fully adaptive, with the full ml degress of freedom, but requires only 1 feedback delay line and one additional delay line per adaptively formed beam. This is a significantly reduced hardware complexity over conventional beamforming algorithms that use time delay and sum (TDS) adaptive networks that require m input tapped delay lines. The massively parallel LMS adaptation of holographically stored adaptive weights results in a processing throughput that can track the real-time adaptive requirements of the sonobuoy array processing problem.

4 Milestones and Accomplishments

Phase 1 First 9 months (plus 6 month no cost extension)

- *Develop initial theoretical model to show the equivalence of the beam former and null-steerer to a space-time Weiner filter.*

  The theoretical model showing the equivalence of the BEAMTAP approach to an optimal space-time adaptive filter is presented in this report, and will be submitted for publication shortly.

- *Use the theoretical model to analyze the frequency response and dynamics of the BEAM-TAP architecture.*

  A comprehensive numerical modell evaluating the suitability of BEAMTAP array processing for sonar applications was developed in MATLAB. The adaptation of the fre-

quency response of a sonar BEAMTAP system implementing adaptive beamforming is illustrated with the computer simulations shown in Figure 4. The dynamics of this adaptation are illustrated in Figure 5, which shows that the SNR improvement of the BEAMTAP sonar array processing system achieves nearly optimal behavior (within a few dB of the ideal system).

- *Complete design of nullsteering optical processor.*

  The design of a null steering optical processor has been presented in various papers and is summarized in this report and is illustrated in Figure 3.

- *Develop modulation scheme for simulating sonar array.*

  We developed a scheme to simulate the sonar modulated light emitted from an array of fibers through using a stroboscopically illuminated acoustooptic device driven by an arbitrary waveform generator. In addition, the variable optical magnification of the acoustooptic device could be used to change the effective angle of arrival of plane wave sonar fields on the array, and zoom lenses were investigated for this purpose. This scheme was designed but was not implemented during Phase 1 of this effort.

- *Develop and test tap-in and tap-out delay lines.*

  We developed and tested a TDI CCD for application in sonar bandwidth signal processing as the tap-in detector technology. The device we tested had difficulty in slowing down the clock below 100KHz so time delays of only 20ms were available in this 2048 pixel device. This range would be sufficient for anything but the largest sonar arrays.

- *Consider alternate algorithms for possible optical implementation.*

We have considered LMS and P-vector adaptation for the sonar problem extensively. In addition we have begun to investigate finite-impulse response (FIR) neural networks[5] for both implementing the array processing functions as well as for performing pattern recognition on the beamformed signals.

Year 2 The option for the second phase of funding for this project was not exercised, so the following milestones were not completed.

- *Obtain all parts for beamsteering demonstration system.*
- *Complete demonstration of efficient TTD optical beamforming.*
- *Investigate all-optical approaches that do not require electronic subtraction nodes.*
- *Modify numerical model to include device specific effects such as incoherent erasure and noise.*
- *Compare the theoretical model, computer simulations, and experimental results.*
- *Submit final report that summarizes these results.*

  This report summarizes our results.

5 Technical Overview

Broadband true-time-delay (TTD) processing is essential in applications where the bandwidth of sources is large compared to their center frequency. Without suitable broadband processing, sources that emit a range of temporal frequencies may appear to be coming from a range of angles, preventing them from being well discriminated from other sources.

Until now broadband beamforming in the time domain has largely been based on running a tapped delay line (TDL) from every array element, to sample the signal at various time delays. While this approach works fine for digital processing, implementing it optically requires a huge amount of hardware as the number of elements becomes large. The approach we have developed, which we call "BEAMTAP" (for Broadband Efficient Adaptive Method for Time-delay Array Processing), replaces any number of tapped delay lines (perhaps 1000 or more with 1000 or more array elements) with only two delay lines: a tap-out delay line for updating the adaptive weights, and a tap-in delay line for integrating the output. These can be implemented optically as a scrolling input modulator and a time-delay-and-integrate output detector. The adaptive weights can then be be optically implemented very compactly (for example by use of a photorefractive crystal, which requires less than a cubic centimeter of space), reducing the costs associated with space, hardware, and power consumption [1].

This architecture thus allows us to exploit the potential advantages of optical processing for broadband beamforming: (1) efficient interfacing with front-end optical sensors or optical postprocessors, and (2) parallel architecture that can accommodate the processing throughputs required for real time adaptation when the number of weights becomes prohibitively large. The main drawback of this architecture appears to be that it requires a bulk delay to present the processor with a delayed version of the array input, used for writing the weights. In this paper we will show the equivalence of this approach to the conventional approach, both analytically and with the results of computer simulations. We will show how it can be implemented in both active scenarios (where there is a known signal of interest) and passive listening scenarios (where the "P-Vector" algorithm can be used to focus on a direction of interest).

6 LMS algorithms: Conventional TTD and BEAM-TAP

LMS is a gradient descent algorithm governing the evolution of a set of weights which filter the array input to produce an output [6]:

$$o(t) = \underline{W}^{\dagger}(t)\underline{Y}(t) \tag{1}$$

In narrowband processing, $\underline{Y}$ is a vector containing the input signals from each sensor in the array. The weight vector, $\underline{W}$, acts as a spatial filter/beamformer and its length is also determined by the number of array elements or sensors, M. A measure of the performance of these weights is the error between the actual output and some desired signal of interest, $\epsilon(t) = d(t) - o(t)$. Consider the gradient of the *average* magnitude-squared of this error, $E[|\epsilon(t)|^2]$:

$$\begin{aligned} \frac{\partial}{\partial \underline{W}} E[|\epsilon(t)|^2] &= E\left[(d - \underline{W}^{\dagger}\underline{Y})(d^* - \underline{Y}^{\dagger}\underline{W})\right] = -E[\underline{Y}(t)\epsilon^*(t)] \\ &= -E[\underline{Y}d^*] + E[\underline{Y}\underline{Y}^{\dagger}]\underline{W}(t) \\ &= -\underline{P}_{Yd} + \underline{\underline{R}}_{YY}\underline{W}(t) \end{aligned} \tag{2}$$

(See [7, 8] for discussions on taking gradients of complex forms). The gradient is taken with respect to the expected error because the error will vary at different times under different manifestations of the signal. $\underline{P}_{Yd}$, called the "P-Vector", is the cross-correlation between the array input and the desired signal, and $\underline{\underline{R}}_{YY}$ contains the cross-correlations between array sensors. The optimal weights that minimize the mean-square-error are obtained when this gradient is zero:

$$\underline{W}_{opt} = \underline{\underline{R}}_{YY}^{-1}\underline{P}_{Yd} \tag{3}$$

When the number of array sensors and taps gets large, an iterative method can be used to generate these weights without calculating the matrix inverse $\underline{\underline{R}}_{YY}^{-1}$ directly. The LMS approach is to modify the weights according to the gradient of the *instantaneous* error [6]:

$$\begin{aligned} \frac{\partial \underline{W}}{\partial t} &= -\mu \frac{\partial}{\partial \underline{W}} |\epsilon(t)|^2 = +\mu \underline{Y}(t)\epsilon^*(t) \\ &= +\mu \underline{Y}(t)d^*(t) - \mu \underline{Y}(t)\underline{Y}^\dagger(t)\underline{W}(t) \end{aligned} \tag{4}$$

This update rule works under the assumption of *slow adaptation*, meaning that the weights evolve more slowly than the time scales on which the array input signals change. This condition will be satisfied if $\mu \ll \frac{B}{\lambda_{max}}$, where B is the bandwidth of the array input and λ_{max} is the largest eigenvalue of $\underline{\underline{R}}_{YY}$. Under these conditions, the weights are found to converge to the same optimal expression given in Equation 3.

The above rule can be implemented in situations where a "desired" signal which correlates well with a source of interest is available, such as in active radar. However, even when an explicit waveform is not available, the cross-correlation vector $\underline{P}_{Yd}$ can still be estimated from the signal bandwidth and angle of arrival. The P-Vector algorithm, originally suggested by Griffiths [9], uses the P-Vector instead of a desired signal to update the weights:

$$\frac{\partial \underline{W}}{\partial t} = +\mu \underline{P}_{Yd} - \mu \underline{Y}(t)\underline{Y}^\dagger(t)\underline{W}(t) \tag{5}$$

This approach is more appropriate for the passive radar/sonar case, where a signal waveform is not known *a priori*, but its bandwidth and a desired look direction can be specified.

Separating broadband signals requires temporal as well as spatial processing. In conventional TTD processing (Figure 2) the signal from each array sensor is sampled at L time delays and multiplied by a weight, thus implementing a Finite Impulse Response (FIR) filter

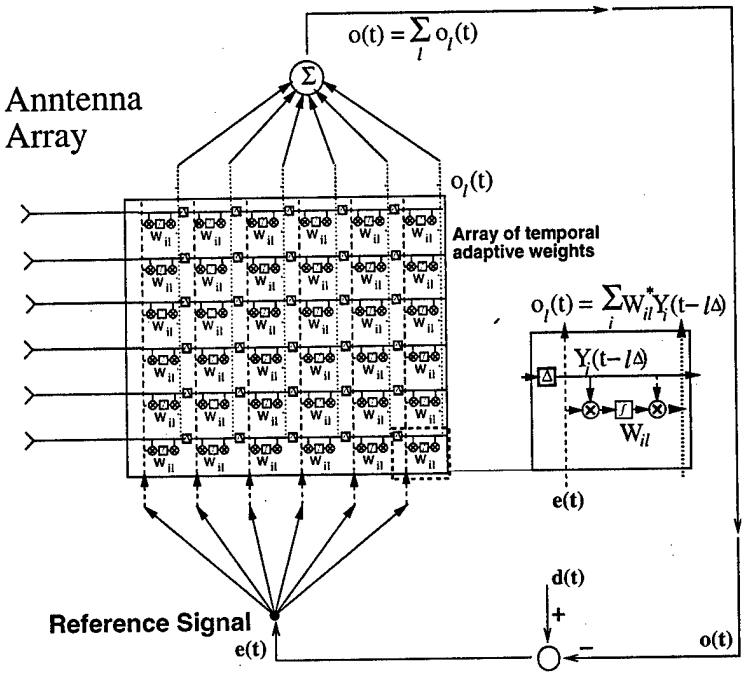

Figure 2: The conventional approach to implementing broad band beamforming with time delay. Weights multiply not only the current signal, but L delayed versions of the signal. This is implemented with tapped delay lines that delay the signal coming from every array sensor.

for every array element. Here it is the array input that is successively delayed to update and read out the weights, whereas the feedback signal affects all the weights simultaneously. The weights can be viewed as a matrix with M rows which act as temporal filters and L columns which act as spatial filters. Alternatively, to use the notation in the equations above, the columns of this weight matrix can be stacked up to form a vector of length $M \times L$. Similarly, $\underline{Y}$ is then also an $M \times L$ vector containing the delayed versions of the array input. (Except in Section 8, where we look in detail at the "spatial filters" associated with each time delay, this is the notation that will be assumed.)

Figure 3 illustrates the algorithm we have developed for optical implementation, BEAM-TAP (Broadband Efficient Adaptive Method for Time-delay Array Processing) [4], which

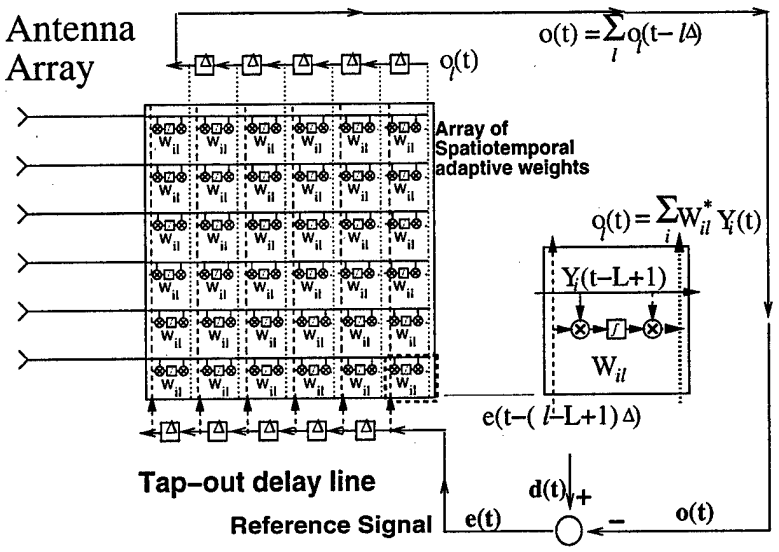

Figure 3: Broadband Efficient Adaptive Method for TTD Array Processing (BEAMTAP): Instead of a delay line from every array sensor, there are just two delay lines: one to delay the feedback signal which writes the weights, and another to read out the weights by delaying the column outputs before they are summed up to form the final output.

instead uses only two delay lines. In this case the array input simultaneously affects all the weights, whereas it is now the feedback signal that is delayed to update them. The tap-out delay line at the bottom delays this feedback, causing the weights to be updated by the current array input and the feedback signal at successive delays. The tap-in delay line at the top is used for reading out the weights (thus filtering the array input). It successively delays the result of multiplication by each column of weights before summing up the column outputs to form the final output. The details of the BEAMTAP algorithm are discussed in Section 8, where its equivalence with conventional time-delay beamforming is established.

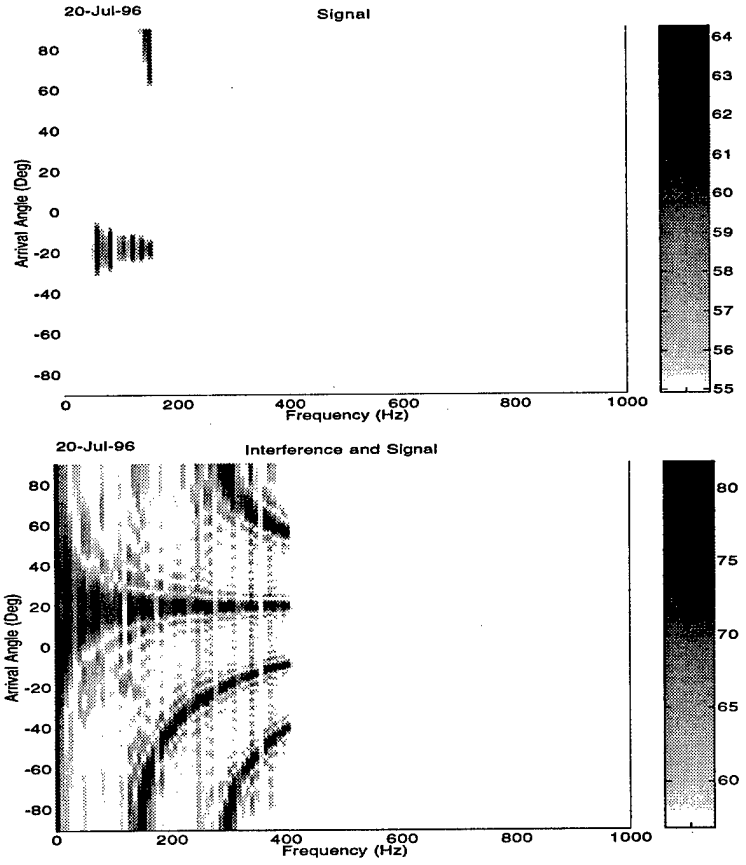

Figure 4: The top plot shows the power distribution of the signal, mapped into frequency and angle. The bottom plot shows the superposition of both the signal and the interference, which is 20 dB higher. The signal is now barely visible at -20 degrees. Above 100 Hz, there is aliasing into other angles because of the array spacing.

7 Computer Simulations

We have conducted computer simulations that verify the equivalence to the BEAMTAP algorithm to conventional TTD. The signal and interference fields in these simulations are relatively simple far field sources, with angles of incidence at -20 and +20 degrees. The fields are broadband with an assumed frequency scaling appropriate for sonar signals: a flat signal

spectrum from 50 to 150 Hz., and a flat interference spectrum from 0 to 400 Hz. The power level for the interferer has been set at 20 dB above the signal. In addition there is white noise, uncorrelated between array sensors, at the same power level as the signal.

The power distribution of the sources, in angle of arrival and frequency, is shown in Figure 4. [The simulated data for all the elements was stacked into a 2-D data matrix. The spectral estimate was obtained by taking the fourier transforms of the rows (the time signals) followed by scaled fourier transforms of the columns to map out the angles of arrival (only simple rectangular windows were used).]. The array was assumed to be linearly spaced, with a spacing equal to half the wavelength of a 100 Hz signal. Higher frequency (and thus smaller wavelength) signals are "spatially aliased" into other angles in the power plot. This is an artifact of the array being inadequately spaced to distinguish signals with higher spatial frequency.

Both the conventional TTD and BEAMTAP adaptive LMS algorithms were applied to the data and the results compared. In both cases, it was the P-Vector version of the algorithms that was implemented. To decrease computing time, the number of array sensors was set to only 10. The evolution of the signal-to-noise ratio is shown in Figure 5. To determine SNR, the signal is defined as the expectation of the signal power filtered through the instantaneous weights, averaged over different realizations of the signal; the noise is similarly defined:

$$SNR(t) = \frac{E[|\underline{W}^{\dagger}(t)\underline{S}(t')|^2]}{E[|\underline{W}^{\dagger}(t)\underline{N}(t')|^2],} = \frac{\underline{W}^{\dagger}(t)\underline{\underline{R}}_{SS}\underline{W}(t)}{\underline{W}^{\dagger}(t)\underline{\underline{R}}_{NN}\underline{W}(t)} \tag{6}$$

Here the averaging is performed over t', $\underline{S}$ is the uncorrupted signal vector, and $\underline{N}$ includes the white noise and interference. Figure 5 shows nearly indentical SNR improvement with the two algorithms. Both approach the optimal SNR, shown by the horizontal line, that can

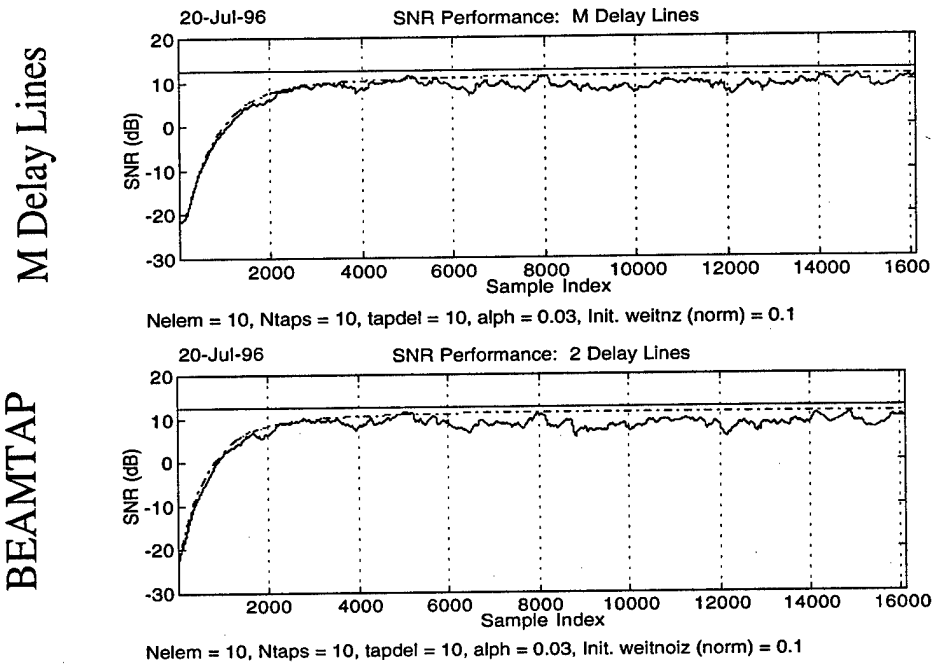

Figure 5: The SNR improvement over time with the conventional and BEAMTAP algorithms (solid line ——). The SNR given by the optimal weights of Equation 3 is the straight line. The SNR of the estimated mean weights (averaged over the inverse bandwidth of the signal) is also shown (dashed line - - - -).

be achieved by correlation matrix inversion (Equation 3). (The update rate parameter μ was set at $\approx 0.025\frac{1000Hz}{tr(\underline{\underline{R}}_{YY})}$).

Another measure of performance is the array gain pattern, which shows the sensitivity of the array vs. angle-of-arrival and frequency. Figure 6 shows the beam-patterns for two sets of weights. The top plot shows the response of the P-Vector weights, which contain information about the signal, but have no information about the interference. Thus there are undesirable sidelobes of sensitivity at 20 degrees, the angle-of-arrival of the interference. The bottom plot shows the gain pattern of weights adapted with the BEAMTAP feedback algorithm. The sidelobes at 20 degrees have been suppressed. Also, this null at 20 degrees

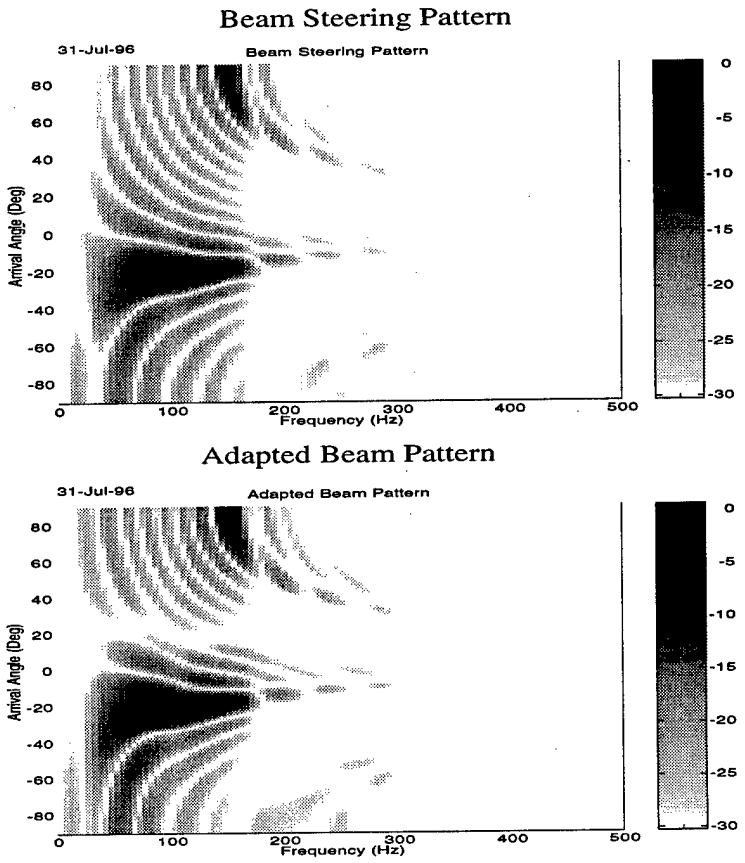

Figure 6: The response of the P-Vector weights (top) and the adapted weights (bottom) as a function of angle and frequency. Note that the adapted weights dig a squint free null at the angle of arrival of the interferer, 20 degrees.

does not shift with frequency (i.e., it is "squint" free).

8 Equivalence of BEAMTAP to Conventional TTD Beamforming

We now present a more detailed discussion of the conventional TTD and BEAMTAP algorithms, and an analysis to establish their correspondence. Here our notation will distin-

guish delayed versions of the array input, $\underline{Y}(t)$.

8.1 The Conventional Algorithm

The conventional algorithm employs a tapped delay at each of the M sensors in the array (Figure 2), producing L delayed versions of the array input, $\underline{Y}(t - l\Delta)$, where Δ is the delay between taps. Each delayed version of the array input is then multiplied by a column of weights, $\underline{W}_l(t)$. (Instead of considering one weight vector of length $M \times L$, in this section we are considering separately the L vectors of length M.) The weighted inputs are all summed to form the output:

$$o(t) = \sum_{l=0}^{L-1} \underline{W}_l^{\dagger}(t)\underline{Y}(t - l\Delta). \tag{7}$$

If a desired signal is known and used to generate an error, $\epsilon(t) = d(t) - o(t)$, the weights can be modified by the gradient of this error with respect to the weights:

$$\frac{\partial \underline{W}_l}{\partial t} = -\mu \frac{\partial}{\partial \underline{W}_l} |\epsilon(t)|^2 \tag{8}$$

By evaluating the gradient in a similar manner as that used to obtain Equation 2 [7, 8], we find an update rule for the weights in terms of the fed back error:

$$\frac{\partial \underline{W}_l}{\partial t} = \mu \underline{Y}(t - l\Delta)\epsilon^*(t) \tag{9}$$

Note that it is the same delayed versions of the array input which read out the weights in Equation 7, $\underline{Y}(t - l\Delta)$, that also help *write* the weights with the current error signal.

Alternatively the P-Vector, $\underline{P}_{Yd,l} = E[\underline{Y}(t - l\Delta)d^*(t)]$, and the current output can be used to update the weights:

$$\frac{\partial \underline{W}_l}{\partial t} = \mu \underline{P}_{Yd,l} - \mu \underline{Y}(t - l\Delta)o^*(t) \tag{10}$$

8.2 The BEAMTAP Algorithm

In BEAMTAP, the current array input is multiplied by all of the weights simultaneously. However, the output from each column of weights, $o_l(t)$ is delayed successively before being summed to form the final output, as shown in Figure 3. This means that the final output is the sum of delayed products of the array signals and weights:

$$o(t) = \sum_{l=0}^{L-1} o_l(t - l\Delta) = \sum_{l=0}^{L-1} \underline{W}_l^\dagger(t - l\Delta)\underline{Y}(t - l\Delta) \quad (11)$$

$$= \sum_{l=0}^{L-1} \underline{H}_l^\dagger(t)\underline{Y}(t - l\Delta) \quad (12)$$

Two symbols have been used here for the weights: $\underline{W}_l^\dagger(t)$ are the weights at the current time, while $\underline{H}_l^\dagger(t)$ are the *effective* weights that determine the current output, as in Equation 7. These effective weights are past values for $l > 0$.

In the BEAMTAP architecture, the current weights are adapted with a feedback signal that is delayed before updating each column of weights. The weights are adjusted with the product of this signal (in this case, the error) and the array input:

$$\frac{\partial \underline{W}_l}{\partial t} = \mu \underline{Y}(t - \tau)\epsilon^*(t - \tau + l\Delta) \quad (13)$$

(Here, the array input has been subjected to a bulk delay of $\tau = (L-1)\Delta$, which is the time it takes a signal to propagate the entire delay line length. This bulk delay is discussed below.) Substituting the relation between the current and effective weights, $\underline{H}_l(t) = \underline{W}_l(t - l\Delta)$, we obtain the following update rule for the effective weights:

$$\frac{\partial \underline{H}_l}{\partial t} = \mu \underline{Y}(t - \tau - l\Delta)\epsilon^*(t - \tau) \quad (14)$$

By comparing Equations 7 and 12, and Equations 9 and 14, we can see that the BEAMTAP algorithm is almost completely equivalent to the conventional one. The only difference

is that the product which updates the weights is delayed by τ, as given in Equation 14. This difference should be negligible for stable adaptation rates, which are typically slow compared to τ.

In order to achieve this equivalence, the array input used to write or update the weights needs to be delayed by τ from the input used to read out or evaluate the weights. This requires that a *bulk* delay be used to present two versions of the array input to the processor. This should be compared with the conventional algorithm, which requires a *tapped* delay line for every array sensor.

In the P-Vector variant, the current weights are adapted by the P-Vector and the fed back output:

$$\frac{\partial \underline{W}_l}{\partial t} = \mu \underline{P}_{Yd,l} - \mu \underline{Y}(t-\tau) o^*(t-\tau+l\Delta) \tag{15}$$

This gives a corresponding update rule for the effective weights:

$$\frac{\partial \underline{H}_l}{\partial t} = \mu \underline{P}_{Yd,l} - \mu \underline{Y}(t-\tau-l\Delta) o^*(t-\tau) \tag{16}$$

Again, this can be compared with Equation 10, with the only difference being a total delay of τ in the update term.

9 Conclusion

We have presented an alternative algorithm to conventional true-time-delay broadband adaptive array processing, BEAMTAP. By reducing the required number of delay lines from M, the number of array elements, to only two, it suggests a more attractive optical hardware implementation, thus making it more feasible to take advantage of the potential benefits of

optical processing, for real-time parallel processing of a large number of adaptive weights, and for efficient coupling to optical sensors or post processors. By separating the delay lines from the adaptive weights, the BEAMTAP algorithm allows the weight multiplications to be readily implemented in a real-time optical storage media, such a photorefractive crystal.

We have shown that analytic equivalence between the two algorithms requires two versions of the array input be presented to the processor, separated by a bulk delay. This also translates to the effective feedback/array input product that updates the weights being delayed; however, this delay should not be significant under typical update rates used for convergence. Similar convergence behavior of the two algorithms was observed under simulations. The new algorithm requires a bulk delay for every array element, rather than a tapped delay line. While still a hardware savings, the bulk delay requirement means that BEAMTAP is probably better suited to higher frequency applications in radar than to sonar. Simulations also verify the ability of the BEAMTAP algorithm to dig an interference null in the beam pattern over a broad frequency range (i.e., a null that is "squint free").

References

[1] R T Weverka, K Wagner, and A Sarto. Photorefractive processing for large adaptive phased-arrays. *Applied Optics*, 35:1344–1366, 1996.

[2] Anthony W. Sarto, Robert T. Weverka, and Kelvin Wagner. Active beam-steering photorefractive phased-array radar processor. In Brian M. Hendrickson, editor, *Optoelectronic Signal Processing for Phased-Array Antennas IV*, volume 2155. SPIE, Jan. 1994.

[3] A. W. Sarto, K. H. Wagner, R. T. Weverka, S. Blair, and S. Weaver. Photorefractive phased array, radar beamforming processor. In *Proc. SPIE*, volume 2845, Denver, CO, August 1996. SPIE.

[4] Kelvin Wagner, S Kraut, L Griffiths, S Weaver, R T Weverka, and A W Sarto. Efficient true-time-delay adaptive-array processing. In *Proc. SPIE*, volume 2845, Denver, CO, August 1996. SPIE.

[5] Paulo E. X. Silveira and Kelvin H. Wagner. Time delay optical neural network. In *Optical Computing Topical Meeting*, June 1998.

[6] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode. Adaptive antenna systems. *Proceedings of the IEEE*, 55(12):2143, 1967.

[7] D. Johnson and D. Dudgeon. *Array Signal Processing: Concepts and Techniques*, pages 402–403,499–502. Prentice Hall, 1993.

[8] B.A. D.H. Brandwood. A complex gradient operator and its application in adaptive array theory. IEE Proc., Pts. F and H, 130(1):11–16, Feb 1983.

[9] L. J. Griffiths. A simple adaptive algorithm for real-time processing in antenna arrays. *Proceedings of the IEEE*, 57(10):1696, 1969.